Railway Scheduling with Hierarchical Reinforcement Learning on a Graph

Thomas Dubach Florian Fuchs Francesco Corman

Institute for Transport Planning and Systems, ETH Zürich, Switzerland

1 Introduction

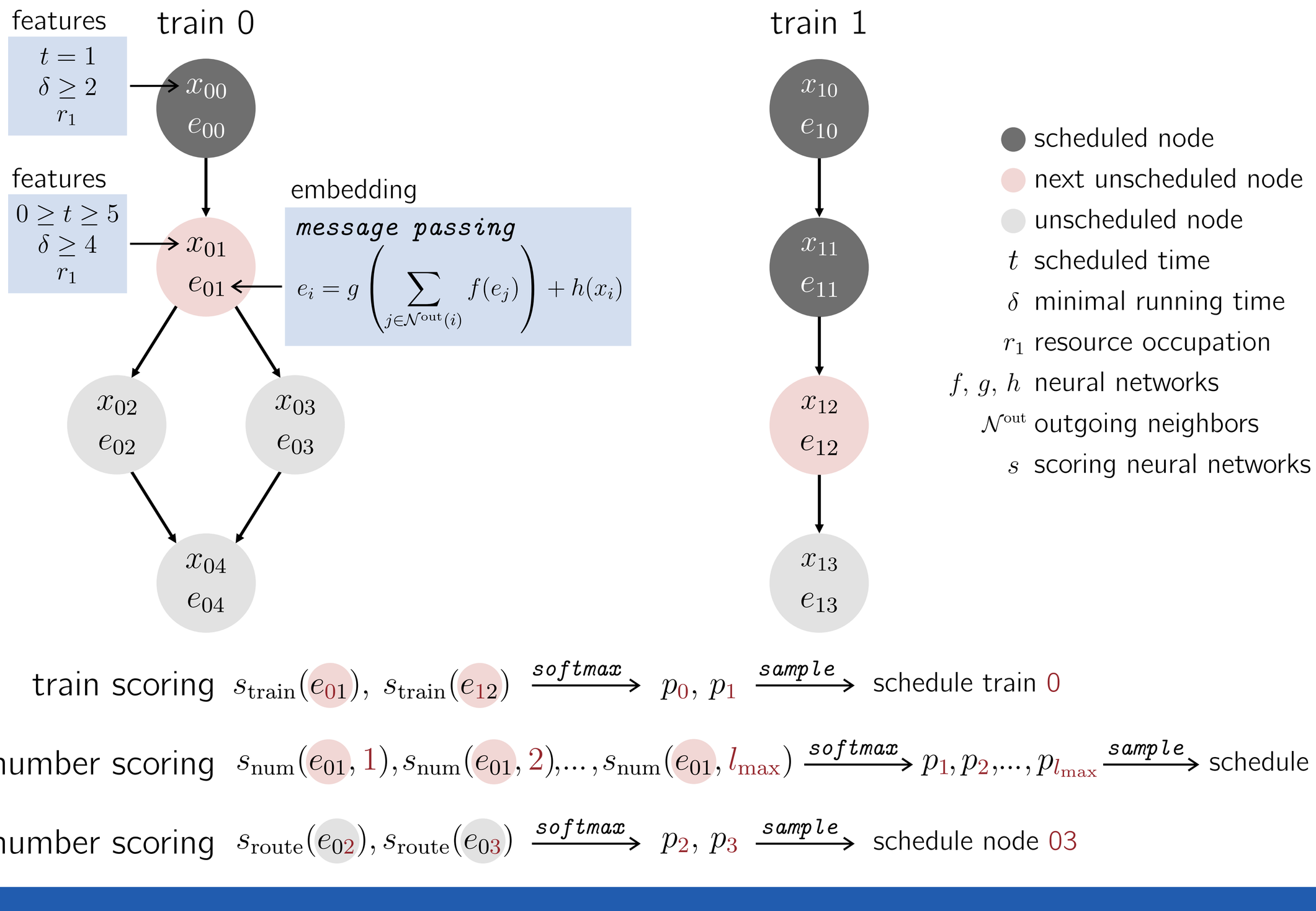
Railway scheduling is a complex task that involves solving recurring instances of constrained optimization problems. Traditionally, each instance is treated as an isolated problem, leading to inefficiencies in computational effort and scalability. This research explores reinforcement learning for optimizing railway timetables based on a graph representation of the scheduling problem.

The graph has nodes which represent operations which have a minimum running time δ, lower and upper bounds for the scheduled time t and a resource which is occupied by the train between $[t,\ t\ +\ \delta]$. The links represent dependencies where exactly one outgoing node must follow immediately after an operation finishes. Thus multiple outgoing links represent alternative routes for a train.

2 Graph embedding & reinforcement learning

We represent the graph state for the RL agent using a graph neural network. A feature vector x_i collects the information on each node i. Via message passing the features on each node are aggregated with the embeddings from its outgoing neighbors (i.e. messages are propagated bottom-up).

The RL agent uses scoring functions to decide which train to schedule next, how many of its nodes to schedule, and which route to take at junctions. These functions take the node embeddings as input and output probability distributions over the respective action spaces. By sampling from these distributions, the agent thus selects a subset of nodes to schedule. Standard RL algorithms such as Actor-Critic update both the scoring functions and the embeddings during training.



3 Reward based on feasibility

Given a set of scheduled nodes and implied precedences between trains, a timetable feasibility solver finds a feasible solution for the remaining unscheduled part. This feasible solution admits an objective value. As reward for the RL agent, we use the change of the objective value with respect to the solution found in the previous step.

If the feasibility solver does not find any solution, it returns a set of nodes causing the infeasibility. These can be traced back to decisions taken by the agent which are penalized.

4 Open questions

- How to represent precedences between trains in the graph?
- How to integrate the feasible solution in the environment state?

5 Selected references

Mao, H., Schwarzkopf, M., Venkatakrishnan, S. B., Meng, Z., & Alizadeh, M. (2019). Learning scheduling algorithms for data processing clusters.

Grinsztajn, N. (2023). Reinforcement learning for combinatorial optimization: Leveraging uncertainty, structure and priors [Université de Lille].

Institut für Verkehrsplanung und Transportsysteme
Institute for Transport Planning and Systems

tdubach@ethz.ch
https://tduba.ch